MQTT and ROC Based Hybrid Robot as a Service (RaaS) Platform

Pritesh C. Bhavsar
Advisor: Prof. Sarosh H. Patel
Department of Computer Science, University of Bridgeport, Bridgeport

ABSTRACT

Robots are rapidly evolving from factory work-horses to robot-companions. The future of robots will be as companions in the workplace functioning as interactive salespeople. In order to support this transition, it is important to combine service-oriented architecture and robotics. Service-oriented architecture and cloud computing have become dominant computing paradigms, and adding an RaaS (Robot as a Service) unit as a part of this system will help the companies manage and develop robots more efficiently. The major components of RaaS will be the integration of RMS (Robot Management System) and ROC (Robot Operation Center). As robots are increasing in the service industry, the inter-robot communication is very critical.

Enhanced RaaS Architecture

Figure 1 depicts the overall architecture of the RaaS Cloud Platform. With the inception of this architecture, we have improved on service interfacing and communication channels by adapting Microservices paradigm and introducing web hooks, MQTT REST bridge and COaP communication channel.

The whole purpose of adapting to the microservice paradigm enables developers to program the specialized service independently as all the services are loosely coupled. With our continuous experimentation on the architecture we soon realized we need to incorporate further protocol support for COaP and RESTful API. In order to enable this we adapted to MQTT-REST bridge.

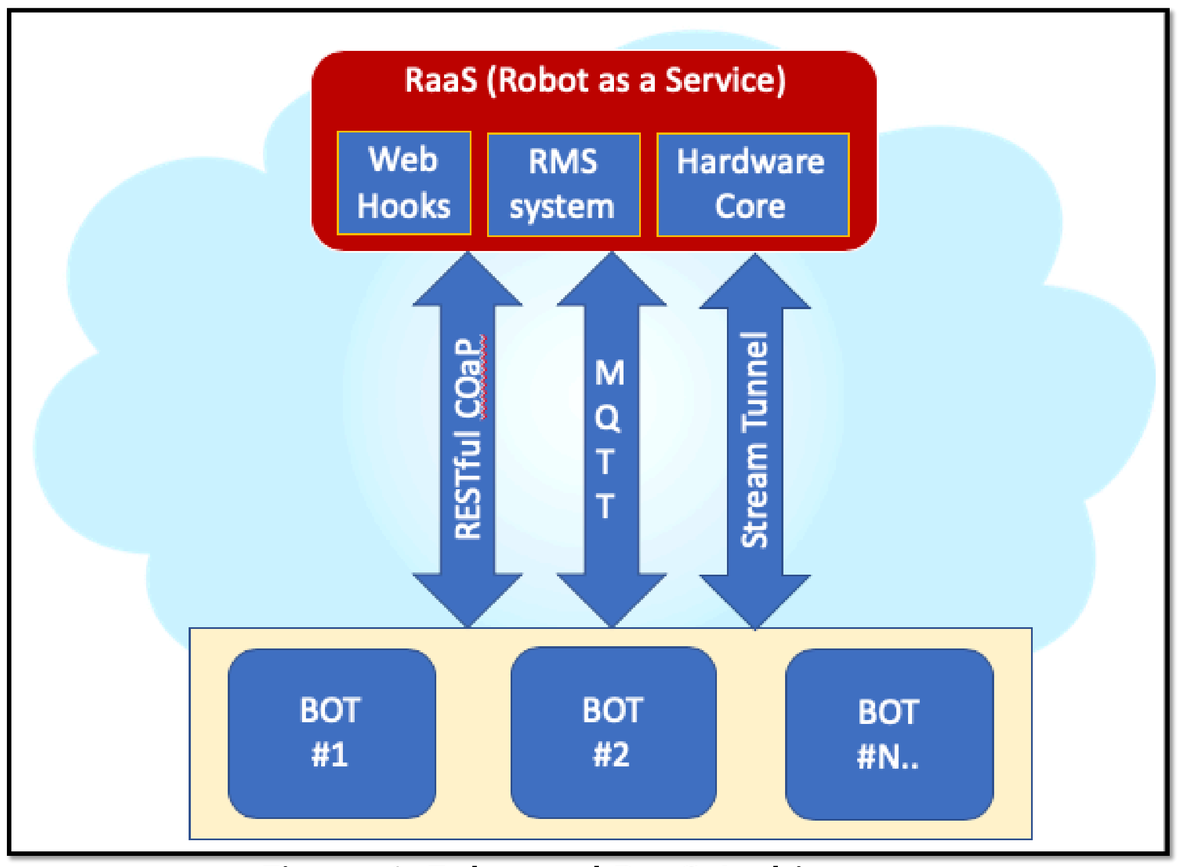


Figure 1: Enhanced RaaS Architecture

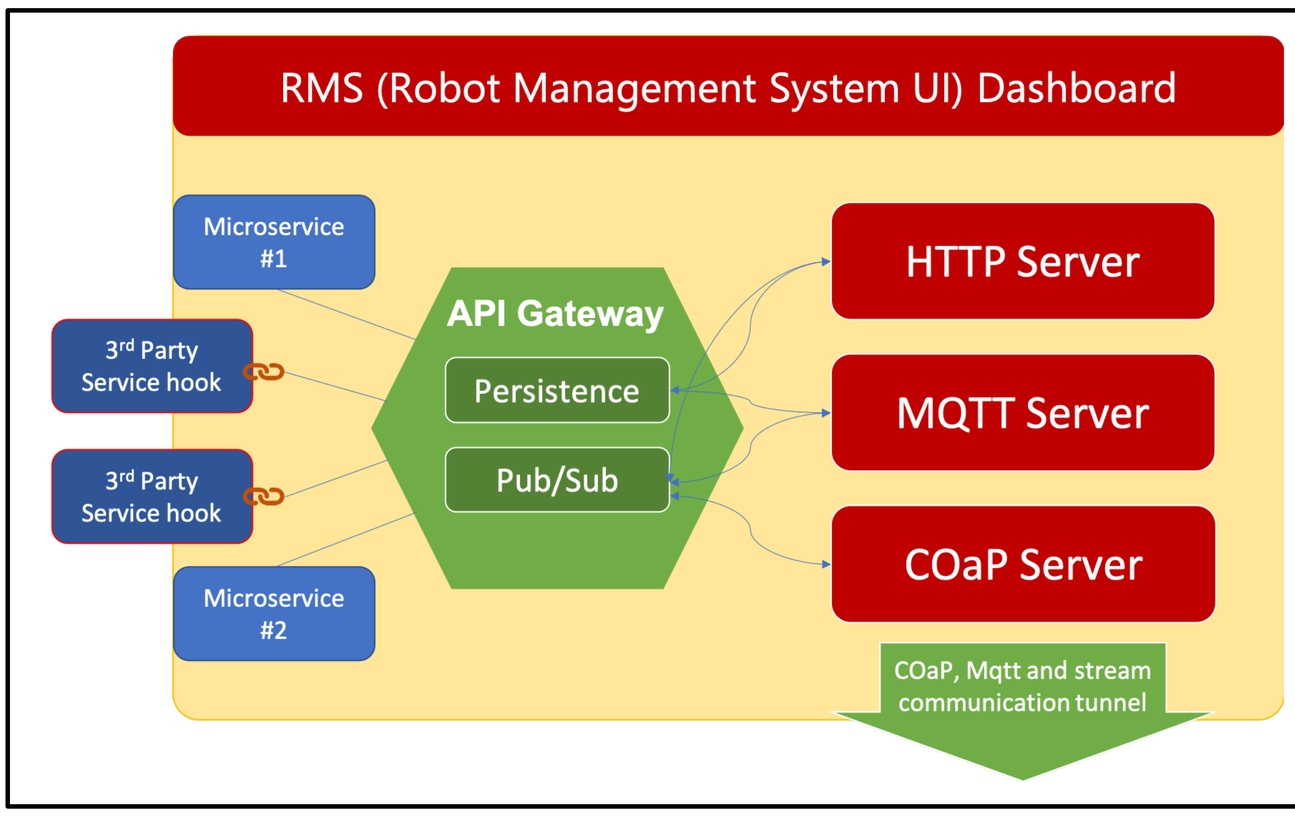


Figure 2: Enhanced Service Core

Cloud Service Core Enhancements

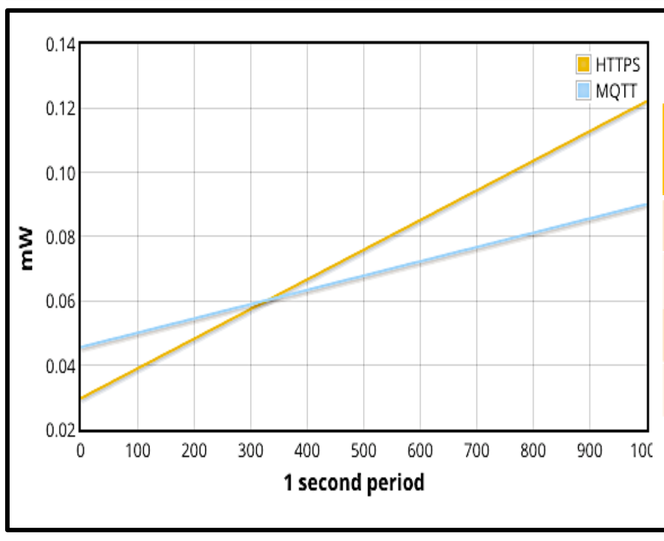
Figure 2 focuses on the core service architecture. The API Gateway along with HTTP, MQTT and COaP Servers provides MQTT REST Bridge. The Bridge architecture leverages the Microservice use, as the whole idea of utilizing microservice gives us the flexibility to build specialized services which are loosely coupled and can be written in any programming language. Also we can hook the 3rd party web services with the proper configurations.

The Bridge handles processing of the service request and resource allocation since the API gateway intelligently chooses the underlying server. With this we can enable the bot with any feature, for eg. Speech Recognition, Facial Recognition, Analytical agent etc.

Power Consumption Comparison

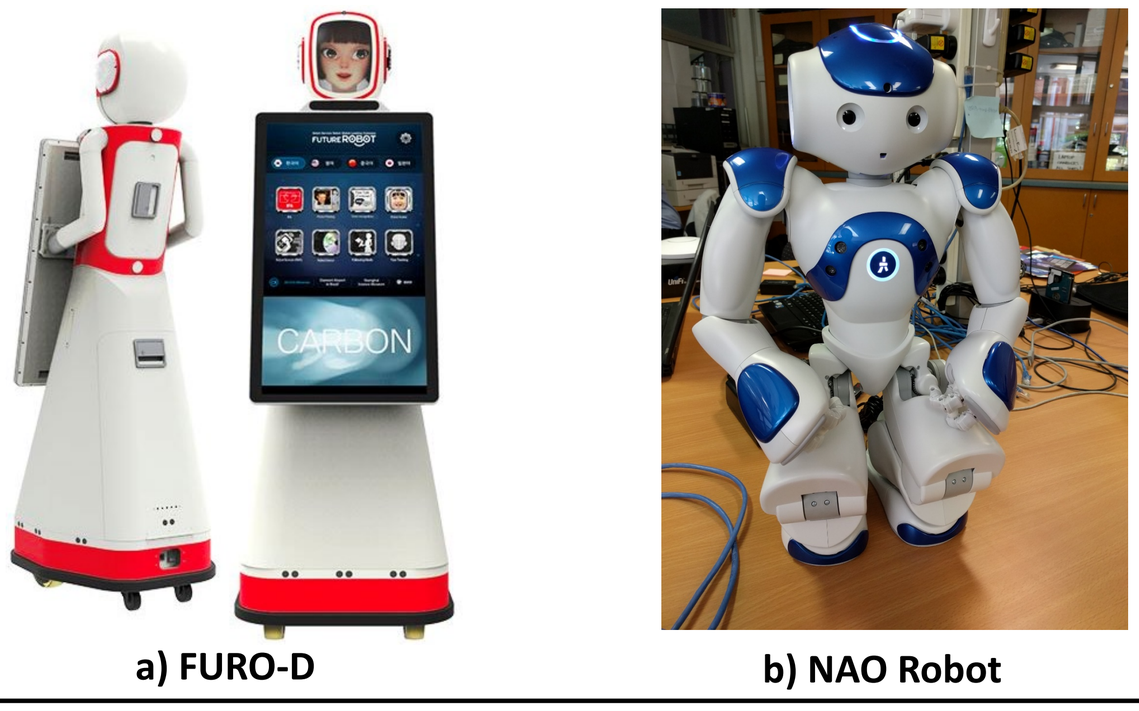
The Figure 3 depicts our experiment on comparison between Legacy HTTPS vs MQTT. As you can see, this is where MQTT gains back ground. In all cases it uses less power, and in most cases a fair bit less. So the longer the connection is established, the 'cheaper' MQTT is to use.

MQTT requires bit more initial power while establishing the connection. After that the power consumption is very less.



Keep Alive (Seconds)	HTTPS	MQTT
60	0.15839	0.01055
120	0.08774	0.00478
240	0.02897	0.00230
480	0.00824	0.00112

Figure 3: HTTPS & MQTT Power consumption comparison



a) FURO-D

b) NAO Robot

Figure 4: Testing Robots

Robot Testing Platform

The Figure 4 depicts two robots which are used to test our model Cloud RaaS Platform. For our case study we used FURO-D to demonstrate the platfrom using preexisting AWS cloud services. We also incorporated the Amazon's AWS Robomaker services. We are now testing our platform on NAO Robot, since it has higher Degree of Freedom.

References

1) Costa, Luis F. and Gonc¸alves, Luiz M. G.Universidade Federal do Rio Grande do Norte DCA-CT-UFRN, Campus Universitario, Lagoa Nova, CEP 59.078-970, Natal, RNluis.feliphe.pb@gmail.com, lmarcos@natalnet.br
2) Yuehai Wang, Jianyi Yang, Robot as a Service in Information Science & Electronic Engineering Education, IEEE 2017